Usbkill - A Program for the very Paranoid Computer User

by Aaron Grothe ajgrothe@yahoo.com

One of the first things the authorities or a company will usually do when they grab a computer is to "secure" the computer. This usually involves the following steps: making sure the user cannot touch or do anything else with the computer (such as close the lid of a laptop, unplug the power, or type anything on it), next is usually installing a device called a "Mouse Jiggler", the final step is usually making sure the computer has power to either through battery or a UPS so they can investigate it at their leisure.

A Mouser Jiggler is a simple USB devices that simulate a mouse and jiggles the cursor a few pixels every few seconds. The purpose of these is to prevent your computer from engaging the screen saver or doing anything else it might do while idle, such as unmounting encrypted drives, and so on. There are also similar devices that will emulate a keyboard and hit the shift key in the same manner. These devices are readily available online just look for mouse jiggler.

What can you Do?

On Linux/BSD and Mac OS X there is a program called usbkill which when installed and running on your computer will monitor the USB bus of your system and shutdown the system if it detects any changes to your attached USB devices (adding or removing). In this example once the "Mouse Jiggler" is installed the system will shutdown the system and optionally perform some basic security cleanup (removing files, wiping memory, swap and so on) as well as running any custom commands you'd like.

What can usbkill do for you?

Remove files
Remove directories
Remove the usbkill program (useful if you only encrypt certain directories)
Wipe Swap
Wipe Ram
Custom Commands

Whitelisting a USB Device

If you have a usb device that you regularly plug and unplug from your computer you can add it to the usbkill whitelist. This way it won’t trip the usbkill command. E.g. I plug and unplug my Nokia phone from my Linux box on a daily basis. To add it to the usb whitelist I followed these steps

```
# lsusb
```

find the entry for the Nokia phone

```
Bus 001 Device 016: ID 0421:06fc Nokia Mobile Phones
```

add the “0421:06fc” to the whitelist section of the usbkill.ini file

Note: USB IDs can be cloned, so keep in mind that this is a potential security risk.

A Few Tips

1) You can have a USB memory stick or other device on a lanyard connected to your wrist. That way if you pull it out of the system it will initiate a shutdown.

This is suggested by the author of the usbkill program.

2)usbkill uses the Secure Delete commands so make sure that you have those utilites installed if you want to be able to do file removal, and other commands. You can also modify the usbkill.ini file to use different commands if you'd prefer.

3) usbkill by default uses the fast versions of the Secure Delete Commands. "sdmem -l" instead of "sdmem", "srm -l" instead of "srm", you can enhance the strength of the wipe by removing the "-l"s from the usb.ini for the additional security. Keep in mind these will also slow down the speed at which your computer halts.

4) To test usbkill without shutting down the computer. To make sure you have everything started correctly you can start usbkill with the --no-shut-down command.

5) If you write a program to start usbkill automatically when you start your system you might want to give it a few minutes to let the USB devices be recognized or else you can end up with a machine that refuses to boot. This one is a personal experience issue :-)

6) Rename the usbkill.py program to another name before you run it. This way if a tech savvy person grabs your computer and you have a longer set of shutdown commands they won't see the program running if they do a ps command.

One enhancement for usbkill

The following is one simple enhancement I've added to my version of usbkill. It adds the capability to send a "pkill --signal USR1 -f usbkill" from a terminal to shutdown the system. One issue with this is that the terminal with this command also needs to be running as root. Here is the patch if anybody else would like to apply it.

Patch

```
--- usbkill.py  2015-09-04 09:55:41.000000000 -0500
+++ usbkill_sigusr1.py  2015-09-22 13:36:41.320000000 -0500
@@ -438,9 +438,18 @@
                log(settings, "[INFO] Exiting because exit signal was received")
                sys.exit(0)

+       # Define SIGUSR1 handler
+       def usr_handler(signum, frame):
+               print("\n[INFO] Starting system shutdown because SIGUSR1 was
receieved\n")
+               log(settings, "[INFO] Starting system shutdown because SIGUSR1
signal was received")
+               kill_computer(settings);
+
        # Register handlers for clean exit of program
        for sig in [signal.SIGINT, signal.SIGTERM, signal.SIGQUIT, ]:
                signal.signal(sig, exit_handler)
+
+       # Kill computer if you receive a SIGUSR1
+       signal.signal (signal.SIGUSR1, usr_handler);

        # Start main loop
        loop(settings)
```

Future?

Usbkill is designed to do one thing and does it pretty well. At the github page for it there are several new feature requests. One of the most interesting is the ability to also detect Thurderbolt, ethernet, and firewire changes. Also the ability for a laptop to detect whether it is running on AC or battery power might be useful as well. The source code is pretty small for usbkill and it is pretty well documented so it is easy to customize it to meet your needs.

Summary

As there is "Security in Depth" there is also "Paranoia in Depth". Tools such as usbkill can be useful if you are doing work on your computer and you would like to be able to quickly shutdown your system in the event that someone tries to grab your computer.

References

Github repo for usbkill - https://github.com/hephaest0s/usbkill
Homepage for Secure Delete Utlities - https://www.thc.org/releases.php?q=delete/